Project course: Optimization
The solution of a difficult problem (facility location)

Michael Patriksson

4 September 2009

Location of facilities which serve customers

- Potential sites: $\mathcal{J} = \{1, \ldots, n\}$ (geographical locations)
- Existing customers: $\mathcal{I} = \{1, \ldots, m\}$ (geographical locations)

$f_j =$ fixed cost of using depot j

$c_{ij} =$ transportation cost when customer i's demand is fulfilled entirely from depot j

Decision problem:

- Which depots to open?
- Which depots to serve which customers, and how much?
- **Goal:** minimize cost
- **Assumption:** depots have unlimited capacity (to be removed)

Variables:

$$y_j = \begin{cases} 1, & \text{if depot } j \text{ is set up} \\ 0, & \text{otherwise} \end{cases}$$

x_{ij} = portion of customer i's demand to be delivered from depot j

Uncapacitated facility location (UFL)

$$
\begin{aligned}
z_0^* = \min \quad & \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} c_{ij} x_{ij} + \sum_{j \in \mathcal{J}} f_j y_j & & & (0) \\
\text{s.t.} \quad & \sum_{j \in \mathcal{J}} x_{ij} = 1, & & i \in \mathcal{I} & (1) \\
& x_{ij} - y_j \leq 0, & & i \in \mathcal{I},\ j \in \mathcal{J} & (2) \\
& x_{ij} \in [0, 1], & & i \in \mathcal{I},\ j \in \mathcal{J} & (3) \\
& y_j \in \{0, 1\}, & & j \in \mathcal{J} & (4)
\end{aligned}
$$

(0) Minimize cost

(1) Deliver precisely the demand

(2) Deliver only from open depots

(3) $\boldsymbol{x}$ is the portion of the demand

(4) Do not partially open a depot

Suppose depots have limited capacity

$d_i =$ demand of customer i ($D = \sum_{i \in \mathcal{I}} d_i$)

$b_j =$ capacity of depot j—if it is opened

Constraints:

$$\sum_{i \in \mathcal{I}} d_i x_{ij} \leq b_j y_j, \quad j \in \mathcal{J} \quad (5) \qquad (\Longrightarrow x_{ij} \leq y_j,\ \forall i, j)$$

$\Longrightarrow$ replace (2) with (5)

Capacitated facility location (CFL)

$$
\begin{aligned}
z^* = \min \quad & \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} c_{ij}x_{ij} + \sum_{j\in\mathcal{J}} f_j y_j && && (0)\\
\text{s.t.} \quad & \sum_{j\in\mathcal{J}} x_{ij} = 1, && i\in\mathcal{I} && (1)\\
& \sum_{i\in\mathcal{I}} d_i x_{ij} - b_j y_j \le 0, && j\in\mathcal{J} && (5)\\
& x_{ij} \in [0,1], && i\in\mathcal{I},\ j\in\mathcal{J} && (3)\\
& y_j \in \{0,1\}, && j\in\mathcal{J} && (4)
\end{aligned}
$$

Observation: Total capacity of open depots must cover the entire demand $\Longrightarrow$ an additional (redundant) constraint:

$$(1),(5) \Longrightarrow \overbrace{\sum_{j\in\mathcal{J}} b_j y_j}^{\text{capacity}} \ge \sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{I}} d_i x_{ij} = \sum_{i\in\mathcal{I}} d_i \sum_{j\in\mathcal{J}} x_{ij} = \sum_{i\in\mathcal{I}} d_i\cdot 1 = \overbrace{D}^{\text{demand}}$$

Trick: Exchange x_{ij} for w_{ij} in constraint (1) and in “half” the objective, add the constraints $x_{ij} = w_{ij}$, and let $0 \leq \alpha \leq 1$.

$$
\begin{aligned}
z^* = \min \quad & \alpha \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} c_{ij} x_{ij} \quad + \quad (1-\alpha) \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} c_{ij} w_{ij} \quad + \quad \sum_{j \in \mathcal{J}} f_j y_j \\
\text{s.t.} \quad & \sum_{j \in \mathcal{J}} w_{ij} = 1, \quad i \in \mathcal{I} && (1) \\
& \sum_{i \in \mathcal{I}} d_i x_{ij} - b_j y_j \leq 0, \quad j \in \mathcal{J} && (5) \\
& \sum_{j \in \mathcal{J}} b_j y_j \geq D, && (6) \\
& w_{ij} - x_{ij} = 0, \quad i \in \mathcal{I},\ j \in \mathcal{J} && (7) \\
& x_{ij} \in [0,1], \quad i \in \mathcal{I},\ j \in \mathcal{J} && (3) \\
& w_{ij} \geq 0, \quad i \in \mathcal{I},\ j \in \mathcal{J} && (8) \\
& y_j \in \{0,1\}, \quad j \in \mathcal{J} && (4)
\end{aligned}
$$

- Constraints (7) tie together $(\boldsymbol{x}, \boldsymbol{y})$ with $\boldsymbol{w}$.
- Lagrangian relax these with multipliers λ_{ij}

$\Longrightarrow$ Lagrange function

$$L(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{y}, \boldsymbol{\lambda}) =$$

$$= \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \left[\alpha c_{ij} x_{ij} + (1-\alpha) c_{ij} w_{ij} + \overbrace{\lambda_{ij}(w_{ij} - x_{ij})}^{\text{penalty}} \right] + \sum_{j \in \mathcal{J}} f_j y_j$$

$$= \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} (\alpha c_{ij} - \lambda_{ij}) x_{ij} + \sum_{j \in \mathcal{J}} f_j y_j + \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \left[(1-\alpha) c_{ij} + \lambda_{ij} \right] w_{ij}$$

- Subproblem (for fixed value of $\boldsymbol{\lambda}$):
  Minimize the Lagrange function under constraints (1), (5), (6), (3), (8) & (4).
  Separates into one in $(\boldsymbol{x}, \boldsymbol{y})$ and $|\mathcal{I}|$ in $\boldsymbol{w}$.

Subproblem in $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$
\begin{aligned}
q_{\boldsymbol{xy}}(\boldsymbol{\lambda}) = \min_{x,y} \quad & \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} [\alpha c_{ij} - \lambda_{ij}]\, x_{ij} \quad + \quad \sum_{j\in\mathcal{J}} f_j y_j \\
\text{s.t.} \quad & \sum_{j\in\mathcal{J}} b_j y_j \;\geq\; D, && (6) \\
& \sum_{i\in\mathcal{I}} d_i x_{ij} \;\leq\; b_j y_j, \qquad j \in \mathcal{J} && (5) \\
& x_{ij} \;\in\; [0,1], \qquad i \in \mathcal{I},\ j \in \mathcal{J} && (3) \\
& y_j \;\in\; \{0,1\}, \qquad j \in \mathcal{J} && (4)
\end{aligned}
$$

For every $\boldsymbol{y}$-solution (such that $\sum_{j\in\mathcal{J}} b_j y_j \geq D$) we have:

- If $y_j = 0$ then $x_{ij} = 0$, $i \in \mathcal{I}$
- If $y_j = 1$ then $\sum_{i\in\mathcal{I}} d_i x_{ij} \leq b_j$

Value [in $(\boldsymbol{x}, \boldsymbol{y})$-subproblem] of opening depot j

That is: letting $y_j = 1$ ($|\mathcal{J}|$ continuous knapsack problems)

$$[\text{CKSP}_j] \qquad v_j(\boldsymbol{\lambda}) = f_j + \min_{\boldsymbol{x}} \quad \sum_{i \in \mathcal{I}} [\alpha c_{ij} - \lambda_{ij}]\, x_{ij}$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{I}} d_i x_{ij} \le b_j$$

$$x_{ij} \in [0,1], \qquad i \in \mathcal{I}$$

$\Longrightarrow$ **Projection onto $\boldsymbol{y}$-space** (a 0/1 knapsack problem)

$$[\text{0/1-KSP}] \qquad q_{\boldsymbol{xy}}(\boldsymbol{\lambda}) = \min_{\boldsymbol{y}} \quad \sum_{j \in \mathcal{J}} v_j(\lambda) \cdot y_j$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{J}} b_j y_j \quad \ge \quad D,$$

$$y_j \quad \in \quad \{0,1\}, \quad j \in \mathcal{J}$$

Solving the continuous knapsack problems [CKSP$_j$]

Greedy algorithm:

- Sort $\frac{\alpha c_{ij} - \lambda_{ij}}{d_i} < 0$, $i \in \mathcal{I}$, in increasing order

$\Longrightarrow$ indices $\{i_1, i_2, \ldots, i_m\}$, $m \leq |\mathcal{I}|$.

- If $m = 0$ then $x_{ij} = 0, i \in \mathcal{I}$. Else, let $k = 1$ and:
- Let $x_{i_k j} = \min\{1; b_j - \sum_{s=1}^{k-1} d_i x_{i_s j}\}$ and let $k := k + 1$ until $\sum_{s=1}^{k} d_i x_{i_s j} = b_j$ or $k = m$.
- Solution fulfills $\sum_{i \in \mathcal{I}} d_i x_{ij} = b_j$ and $x_{ij} \in [0, 1]$, $i \in \mathcal{I}$.
- $v_j(\boldsymbol{\lambda}) = f_j + \min \sum_{k=1}^{|\mathcal{I}|} \sum_{j \in \mathcal{J}} [\alpha c_{i_k j} - \lambda_{i_k j}] \, x_{i_k j}$.

Solving $0/1$ knapsack problems

Not polynomial. Solve with Dynamic Programming or Branch & Bound (CPLEX).

Solution:

$y_j(\boldsymbol{\lambda}) \in \{0, 1\}$, $j \in \mathcal{J}$

$x_{ij}(\boldsymbol{\lambda}) = 0$, $i \in \mathcal{I}$, if $y_j(\lambda) = 0$

$x_{ij}(\boldsymbol{\lambda}) = x_{ij}$ by the above, $i \in \mathcal{I}$, if $y_j(\boldsymbol{\lambda}) = 1$

Subproblem in $\boldsymbol{w}$
($|\mathcal{I}|$ semi-assignment problems):

$$[\text{SAP}] \qquad q_{\boldsymbol{w}}(\boldsymbol{\lambda}) = \sum_{i \in \mathcal{I}} \left[\begin{array}{ll} \min\limits_{\boldsymbol{w}} & \sum\limits_{j \in \mathcal{J}} [(1-\alpha)c_{ij} + \lambda_{ij}]\, w_{ij} \\ \text{s.t.} & \sum\limits_{j \in \mathcal{J}} w_{ij} = 1, \quad w_{ij} \geq 0, \quad j \in \mathcal{J} \end{array} \right]$$

Solving semi-assignment problem i

- Find ℓ_i such that $(1-\alpha)c_{i\ell_i} + \lambda_{i\ell_i} = \min\limits_{j \in \mathcal{J}} \{(1-\alpha)c_{ij} + \lambda_{ij}\}$.
- Let $w_{i\ell_i}(\boldsymbol{\lambda}) = 1$, $w_{ij}(\boldsymbol{\lambda}) = 0$, $j \neq \ell_i$.

Value of relaxed problem for fixed value of $\boldsymbol{\lambda}$

$$q(\boldsymbol{\lambda}) = \underbrace{q_{\boldsymbol{xy}}(\boldsymbol{\lambda})}_{\text{difficult}} + \underbrace{q_{\boldsymbol{w}}(\boldsymbol{\lambda})}_{\text{simple}}$$

- Can show that $q(\boldsymbol{\lambda}) \leq q^*$ for all $\boldsymbol{\lambda} \in \mathbb{R}^{|\mathcal{I}| \times |\mathcal{J}|}$ (weak duality)
- λ_{ij} is the penalty for violating $w_{ij} = x_{ij}$
- Find best underestimate of $q^* \iff$ find "optimal" values of penalties λ_{ij}
- That is: $\max_{\boldsymbol{\lambda} \in \mathbb{R}^{|\mathcal{I}| \times |\mathcal{J}|}} q(\boldsymbol{\lambda}) \leq q^*$ (most often $\max_{\boldsymbol{\lambda} \in \mathbb{R}^{|\mathcal{I}| \times |\mathcal{J}|}} q(\boldsymbol{\lambda}) < z^*$, *not* strong duality)

How to find better value of λ_{ij}?

Penalty: $\min \ldots \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \lambda_{ij}(w_{ij} - x_{ij})$

- If $w_{ij}(\boldsymbol{\lambda}) > x_{ij}(\boldsymbol{\lambda}) \Longrightarrow$ Increase value of λ_{ij} (more expensive to violate constraint)
- If $w_{ij}(\boldsymbol{\lambda}) < x_{ij}(\boldsymbol{\lambda}) \Longrightarrow$ Decrease value of λ_{ij} (more expensive to violate constraint)
- Iterative method (subgradient algorithm) to find optimal penalties $\boldsymbol{\lambda}^*$:

  $$\lambda_{ij}^{t+1} = \lambda_{ij}^t + \rho_t \left[w_{ij}(\boldsymbol{\lambda}^t) - x_{ij}(\boldsymbol{\lambda}^t) \right], \qquad t = 0, 1, \ldots$$

  where $\rho_t > 0$ is a step length, decreasing with t
- Use *feasibility heuristic* from every $\left[\boldsymbol{x}(\boldsymbol{\lambda}^t), \boldsymbol{w}(\boldsymbol{\lambda}^t), \boldsymbol{y}(\boldsymbol{\lambda}^t) \right]$ to yield a *feasible solution* to CFL (open more depots, send only from open depots, $\boldsymbol{x} = \boldsymbol{w}, \ldots$). Example: Benders' subproblem!

Example: $|\mathcal{I}| = 4$, $|\mathcal{J}| = 3$, $\alpha = \frac{1}{2}$

$$(c_{ij}) = \begin{bmatrix} 6 & 2 & 4 \\ 2 & 8 & 4 \\ 16 & 2 & 6 \\ 10 & 12 & 4 \end{bmatrix}, (f_j) = \begin{bmatrix} 11 \\ 16 \\ 21 \end{bmatrix}, (d_i) = \begin{bmatrix} 6 \\ 4 \\ 8 \\ 5 \end{bmatrix}, (b_j) = \begin{bmatrix} 12 \\ 10 \\ 13 \end{bmatrix}$$

$$\begin{aligned} q_{\boldsymbol{xy}}(\boldsymbol{\lambda}) = \min \quad & \textstyle\sum_{j=1}^{3} v_j(\boldsymbol{\lambda}) \cdot y_j \\ \text{s.t.} \quad & 12y_1 + 10y_2 + 13y_3 \geq 23 \\ & \boldsymbol{y} \in \{0,1\}^3 \end{aligned}$$

Let $(\lambda_{ij}^t) = \begin{bmatrix} 7 & 0 & 0 \\ 3 & 10 & 2 \\ 5 & 2 & 0 \\ 0 & 7 & 5 \end{bmatrix}$

Observe: implies that $y_3 = 1$ must hold.

$$\underbrace{\ldots \Longrightarrow \ldots}_{\text{(next page)}} \quad q_{\boldsymbol{xy}}(\boldsymbol{\lambda}) = \min \quad 5y_1 + 8.875y_2 + 18y_3$$

$$\text{s.t.} \quad 12y_1 + 10y_2 + 13y_3 \geq 23, \quad \boldsymbol{y} \in \{0,1\}^3$$

$$
\begin{aligned}
v_1(\boldsymbol{\lambda}^t) = 11+ \quad & \min \quad -4x_{11} - 2x_{21} + 3x_{31} + 5x_{41} \\
& \text{s.t.} \quad 6x_{11} + 4x_{21} + 8x_{31} + 5x_{41} \le 12, \quad \boldsymbol{x}_{\cdot 1} \in [0,1]^4 \\
\Longrightarrow \quad & \quad x_{11} = x_{21} = 1,\ x_{31} = x_{41} = 0, \quad v_1(\boldsymbol{\lambda}^t) = 5
\end{aligned}
$$

$$
\begin{aligned}
v_2(\boldsymbol{\lambda}^t) = 16+ \quad & \min \quad x_{12} - 6x_{22} - x_{32} - x_{42} \\
& \text{s.t.} \quad 6x_{12} + 4x_{22} + 8x_{32} + 5x_{42} \le 10, \quad \boldsymbol{x}_{\cdot 2} \in [0,1]^4 \\
\Longrightarrow \quad & \quad x_{22} = x_{42} = 1,\ x_{32} = \frac{1}{8},\ x_{12} = 0, \quad v_2(\boldsymbol{\lambda}^t) = 8.875
\end{aligned}
$$

$$
\begin{aligned}
v_3(\boldsymbol{\lambda}^t) = 21+ \quad & \min \quad 2x_{13} + 0x_{23} + 3x_{33} - 3x_{43} \\
& \text{s.t.} \quad 6x_{13} + 4x_{23} + 8x_{33} + 5x_{43} \le 13, \quad \boldsymbol{x}_{\cdot 3} \in [0,1]^4 \\
\Longrightarrow \quad & \quad x_{23} = x_{43} = 1,\ x_{13} = x_{33} = 0, \quad v_3(\boldsymbol{\lambda}^t) = 18
\end{aligned}
$$

Solution to $(\boldsymbol{x}, \boldsymbol{y})$ problem for $\boldsymbol{\lambda} = \boldsymbol{\lambda}^t$

$$\boldsymbol{y}(\boldsymbol{\lambda}^t) = (1,0,1)^{\mathrm{T}},\ \boldsymbol{x}(\boldsymbol{\lambda}^t) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ q_{\boldsymbol{xy}}(\boldsymbol{\lambda}^t) = 5 + 0 + 18 = 23$$

$\boldsymbol{w}$-problem separates into one for each customer i

$q_{\boldsymbol{w}}(\boldsymbol{\lambda}^t) = \sum_{i=1}^{4} q_{\boldsymbol{w}}^i(\boldsymbol{\lambda}^t)$, where $\left(1 - \alpha = \frac{1}{2}\right)$

$$q_{\boldsymbol{w}}^i(\boldsymbol{\lambda}^t) = \min \quad \sum_{j=1}^{3} \left[(1-\alpha)c_{ij} + \lambda_{ij}^t\right] w_{ij}$$

$$\text{s.t.} \quad \sum_{j=1}^{3} w_{ij} = 1, \quad w_{ij} \geq 0,\ j = 1,2,3$$

$$
\begin{array}{rll}
q^1_{\boldsymbol{w}}(\boldsymbol{\lambda}^t) = & \min & 10w_{11} + w_{12} + 2w_{13} \\
& \text{s.t.} & w_{11} + w_{12} + w_{13} = 1, \quad w_{1j} \geq 0,\ j = 1,2,3 \\
\Longrightarrow & & w_{12}(\boldsymbol{\lambda}^t) = 1,\ w_{11}(\boldsymbol{\lambda}^t) = w_{13}(\boldsymbol{\lambda}^t) = 0,\ q^1_{\boldsymbol{w}}(\boldsymbol{\lambda}^t) = 1 \\
\hline
q^2_{\boldsymbol{w}}(\boldsymbol{\lambda}^t) = & \min & 4w_{21} + 14w_{22} + 4w_{23} \\
& \text{s.t.} & w_{21} + w_{22} + w_{23} = 1, \quad w_{2j} \geq 0,\ j = 1,2,3 \\
\Longrightarrow & & w_{21}(\boldsymbol{\lambda}^t) = 1,\ w_{22}(\boldsymbol{\lambda}^t) = w_{23}(\boldsymbol{\lambda}^t) = 0,\ q^2_{\boldsymbol{w}}(\boldsymbol{\lambda}^t) = 4 \\
\hline
q^3_{\boldsymbol{w}}(\boldsymbol{\lambda}^t) = & \min & 13w_{31} + 3w_{32} + 3w_{33} \\
& \text{s.t.} & w_{31} + w_{32} + w_{33} = 1, \quad w_{3j} \geq 0,\ j = 1,2,3 \\
\Longrightarrow & & w_{32}(\boldsymbol{\lambda}^t) = w_{33}(\boldsymbol{\lambda}^t) = \dfrac{1}{2},\ w_{31}(\boldsymbol{\lambda}^t) = 0,\ q^3_{\boldsymbol{w}}(\boldsymbol{\lambda}^t) = 3 \\
\hline
q^4_{\boldsymbol{w}}(\boldsymbol{\lambda}^t) = & \min & 5w_{41} + 13w_{42} + 7w_{43} \\
& \text{s.t.} & w_{41} + w_{42} + w_{43} = 1, \quad w_{4j} \geq 0,\ j = 1,2,3 \\
\Longrightarrow & & w_{41}(\boldsymbol{\lambda}^t) = 1,\ w_{42}(\boldsymbol{\lambda}^t) = w_{43}(\boldsymbol{\lambda}^t) = 0,\ q^4_{\boldsymbol{w}}(\boldsymbol{\lambda}^t) = 5
\end{array}
$$

Solution to $\boldsymbol{w}$ problem

$$\boldsymbol{w}(\boldsymbol{\lambda}^t) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 1 & 0 & 0 \end{bmatrix}, \; q_{\boldsymbol{w}}(\boldsymbol{\lambda}^t) = 13,$$

$$q(\boldsymbol{\lambda}^t) = q_{\boldsymbol{xy}}(\boldsymbol{\lambda}^t) + q_{\boldsymbol{w}}(\boldsymbol{\lambda}^t) = 35$$

$$\Longrightarrow z^* \geq 35$$

New $\boldsymbol{\lambda}$ vector (e.g., $\rho_t = 8$):

$$\begin{aligned} \boldsymbol{\lambda}^{t+1} &= \boldsymbol{\lambda}^t + \rho_t \left[\boldsymbol{w}(\boldsymbol{\lambda}^t) - \boldsymbol{x}(\boldsymbol{\lambda}^t)\right] \\ &= \begin{bmatrix} 7-\rho_t & \rho_t & 0 \\ 3 & 10 & 2-\rho_t \\ 5 & 2+\frac{\rho_t}{2} & \frac{\rho_t}{2} \\ \rho_t & 7 & 5-\rho_t \end{bmatrix} = \begin{bmatrix} -1 & 8 & 0 \\ 3 & 10 & -6 \\ 5 & 6 & 4 \\ 8 & 7 & -3 \end{bmatrix} \end{aligned}$$

Feasible solution $\Longleftrightarrow$ $\boldsymbol{x}(\boldsymbol{\lambda}^t) = \boldsymbol{w}(\boldsymbol{\lambda}^t)$? No $\Longrightarrow$ Feasibility heuristic

Idea: Open depots given by $\boldsymbol{y}(\boldsymbol{\lambda}^t) \Longrightarrow \boldsymbol{y}^H = \boldsymbol{y}(\boldsymbol{\lambda}^t) = (1, 0, 1)^{\mathrm{T}}$.

Send only from open depots ($y_j^H = 0 \Longrightarrow x_{ij}^H = 0, \forall i$).

Fulfill demand but do not violate capacity restrictions:

$$\text{Let } \boldsymbol{x}^H = \begin{bmatrix} \frac{1}{6} & 0 & \frac{5}{6} \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \Longrightarrow$$

$$z^H = 6 \cdot \frac{1}{6} + 4 \cdot \frac{5}{6} + 2 + 6 + 10 + 11 + 21 = 52 + \frac{1}{3}$$

$\Longrightarrow z^* \in [35, 52 + \frac{1}{3}] = [q(\boldsymbol{\lambda}^t), z^H]$ (not very good interval)

- Choice of step lengths (ρ_t) later (subgradient optimization, convergence to an optimal value of $\boldsymbol{\lambda}$)
- Feasibility heuristics can be made more or less sophisticated
- There are more ways in which to Lagrangian relax *continuous* constraints in an optimization problem
- E.g.: Lagrangian relax (1) *or* (5)
  (with multipliers $\mu_i \in \mathbb{R}$ resp. $\nu_j \in \mathbb{R}_+$) in the original formulation (CFL)

- There are also other methods for solving CFL. Consider for example the fact that for fixed $\boldsymbol{y}$, the remaining problem over $\boldsymbol{x}$ is very simple (a transportation problem). Algorithms can be based on only adjusting $\boldsymbol{y}$, always optimizing over $\boldsymbol{x}$ for each $\boldsymbol{y}$. (We say that we *project* the problem onto the $\boldsymbol{y}$ variables.)
- This is the Benders' subproblem (more on the Benders algorithm later).